

AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

- 1 1. (Currently amended) A method for accessing objects stored outside of
2 main memory in an object-addressed memory hierarchy, comprising:
3 receiving a request to access an object, wherein the request includes an
4 object identifier for the object that is used to reference the object within the
5 object-addressed memory hierarchy, wherein the request to access the object is
6 received at a hardware translator within a central processing unit (CPU) and
7 wherein the translator translates between object identifiers (used to reference
8 objects in an object cache within the CPU) and physical addresses (used to
9 address objects in main memory);
10 using the object identifier to retrieve an object table entry associated with
11 the object, wherein the object table entry associates a given object identifier with a
12 corresponding physical address if the object is in main memory and an external
13 location if the object is not in main memory;
14 examining a valid indicator within the object table entry;
15 if the valid indicator indicates the object is located in main memory, using
16 a physical address in the object table entry to access the object in main memory;
17 and
18 if the valid indicator indicates that the object is not located in main
19 memory, relocating the object into memory from a location outside of memory,
20 and then accessing the object in main memory.

1 2 (Canceled).

1 3. (Currently amended) The method of claim 1-~~claim 2~~,
2 wherein prior to receiving the request at the translator, the request is
3 initially directed to the object cache;
4 wherein if the request causes a hit in the object cache, the object is
5 accessed in the object cache and the request is not sent to the translator; and
6 wherein if the request causes a miss in the object cache, the request is sent
7 to the translator.

1 4. (Original) The method of claim 1, wherein relocating the object into
2 main memory involves using location information from the object table entry to
3 determine the location of the object outside of main memory.

1 5. (Original) The method of claim 4, wherein the location information can
2 include:
3 a secondary storage address for the object;
4 a network address for the object;
5 a uniform (or universal) resource locator (URL) for the object;
6 another (possibly different) object identifier associated with the object; and
7 a physical address for a compressed block of objects containing the object.

1 6. (Original) The method of claim 4, wherein the location information is
2 overloaded into a physical address field in the object table entry.

1 7. (Original) The method of claim 1, wherein relocating the object into
2 main memory involves causing an object fault handler to execute in a central
3 processing unit (CPU) to relocate the object into main memory.

1 8. (Original) The method of claim 1, wherein relocating the object into
2 main memory involves overlapping retrieval of multiple objects into main
3 memory from locations outside of main memory.

1 9. (Original) The method of claim 1, wherein after relocating the object
2 into main memory, the method further comprises:
3 updating the valid indicator to specify that the object is located in main
4 memory; and
5 updating the physical address in the object table entry to specify the
6 location of the object in main memory.

1 10. (Original) The method of claim 1, wherein the object is defined within
2 an object-oriented programming system.

1 11. (Currently amended) An apparatus that facilitates accessing objects
2 stored outside of main memory in an object-addressed memory hierarchy,
3 comprising:
4 a receiving mechanism configured to receive a request to access an object,
5 wherein the request includes an object identifier for the object that is used to
6 reference the object within the object-addressed memory hierarchy, wherein the
7 request to access the object is received at a hardware translator within a central
8 processing unit (CPU) and wherein the translator translates between object
9 identifiers (used to reference objects in an object cache within the CPU) and
10 physical addresses (used to address objects in main memory);
11 a object table lookup mechanism configured to use the object identifier to
12 retrieve an object table entry associated with the object, wherein the object table
13 entry associates a given object identifier with a corresponding physical address if

14 the object is in main memory and an external location if the object is not in main
15 memory;
16 an access mechanism configured to,
17 examine a valid indicator within the object table entry,
18 if the valid indicator indicates the object is located in main
19 memory, to use a physical address in the object table entry to
20 access the object in main memory, and
21 if the valid indicator indicates that the object is not located
22 in main memory, to relocate the object into memory from a
23 location outside of memory, and to access the object in main
24 memory.

1 12 (Canceled).

1 | 13. (Currently amended) The apparatus of claim 11 ~~claim 12~~, further
2 comprising the object cache,
3 wherein prior to receiving the request at the translator, the request is
4 initially directed to the object cache;
5 wherein if the request causes a hit in the object cache, the object is
6 accessed in the object cache and the request is not sent to the translator; and
7 wherein if the request causes a miss in the object cache, the request is sent
8 to the translator.

1 14. (Original) The apparatus of claim 11, wherein while relocating the
2 object into main memory, the access mechanism is configured to use location
3 information from the object table entry to determine the location of the object
4 outside of main memory.

1 15. (Original) The apparatus of claim 14, wherein the location information
2 can include:

3 a secondary storage address for the object;
4 a network address for the object;
5 a uniform (or universal) resource locator (URL) for the object;
6 another (possibly different) object identifier associated with the object; and
7 a physical address for a compressed block of objects containing the object.

1 16. (Original) The apparatus of claim 14, wherein the location information
2 is overloaded into a physical address field in the object table entry.

1 17. (Original) The apparatus of claim 11, wherein while relocating the
2 object into main memory, the access mechanism is configured to cause an object
3 fault handler to execute in a central processing unit (CPU) to relocate the object
4 into main memory.

1 18. (Original) The apparatus of claim 11, wherein while relocating the
2 object into main memory the access mechanism is configured to overlap retrieval
3 of multiple objects into main memory from locations outside of main memory.

1 19. (Original) The apparatus of claim 11, wherein after relocating the
2 object into main memory, the access mechanism is configured to:
3 update the valid indicator to specify that the object is located in main
4 memory; and to
5 update the physical address in the object table entry to specify the location
6 of the object in main memory.

1 20. (Original) The apparatus of claim 11, wherein the object is defined
2 within an object-oriented programming system.

1 21. (Currently amended) A computer system that facilitates accessing
2 objects stored outside of main memory in an object-addressed memory hierarchy,
3 comprising:
4 a processor;
5 a main memory;
6 the object-addressed memory hierarchy;
7 an object cache within the object-addressed memory hierarchy;
8 a translator that translates between object identifiers, used to address
9 | objects in the object cache of a central processing unit (CPU), and physical
10 | addresses, used to address objects in main memory;
11 wherein the translator is configured to receive a request to access an object
12 | after the request misses in the object cache of the CPU, wherein the request
13 | includes an object identifier for the object that is used to reference the object
14 within the object-addressed memory hierarchy, and wherein the object table entry
15 associates a given object identifier with a corresponding physical address if the
16 object is in main memory and an external location if the object is not in main
17 memory;
18 a object table lookup mechanism with the translator configured to use the
19 object identifier to retrieve an object table entry associated with the object; and
20 an access mechanism configured to,
21 examine a valid indicator within the object table entry,
22 if the valid indicator indicates the object is located in main
23 memory, to use a physical address in the object table entry to
24 access the object in main memory, and

25 if the valid indicator indicates that the object is not located in main
26 memory, to relocate the object into memory from a location outside of memory,
27 and to access the object in main memory.